



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/066,036	01/31/2002	Kaihu Chen	063170.7005	3956
5073 7590 06/01/2009				
BAKER BOTTS L.L.P. 2001 ROSS AVENUE SUITE 600 DALLAS, TX 75201-2980				
EXAMINER				
PILLAI, NAMITHA				
ART UNIT		PAPER NUMBER		
2173				
NOTIFICATION DATE		DELIVERY MODE		
06/01/2009		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ptomail1@bakerbotts.com

glenda.orrantia@bakerbotts.com

# Office Action Summary

**Application No.**

10/066,036

**Applicant(s)**

CHEN ET AL.

**Examiner**

NAMITHA PILLAI

**Art Unit**

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 13 March 2009.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 17-38 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 17-38 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SE/US)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Response to Amendment***

1. This Office action is responsive to the Request for Continued Examination (RCE) filed under 37 CFR §1.53(d) on 3/13/09. Applicants have properly set forth the RCE, which has been entered into the application, and an examination on the merits follows herewith. The Examiner acknowledges Applicant's amendments to claims 17, 24, 31, 32, 34 and the addition of new claims 37 and 38. All pending claims have been rejected.

### ***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 31-36 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 31 recites a system but the claim does not disclose hardware elements that are included in this system. The specification does not clearly define the processor as a physical hardware device.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2173

3. Claims 17-32 and 34-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over International Publication Number WO 00/73941 A2 (Yalcinalp et al.), herein referred to as Yalcinalp and U.S. Publication No. 2001/0056460 A1 (Sahota et al.), herein referred to as Sahota.

Referring to claim 17, Yalcinalp discloses a method for dynamically constructing a web page (page 6, lines 3-21). Web pages in a NETSCAPE browser are dynamically created using XSL sheets and XML documents. Yalcinalp discloses receiving a request for a webpage identified by a URL (page 7, lines 6-7). Yalcinalp discloses that the webpage is defined by a plurality of GUI nodes stored in a page registry (Figure 4A and page 10, lines 4-19). Yalcinalp discloses receiving a user identifier representing a user associated with the request (page 7, lines 7-11), where the user identifier includes the client type and client configuration associated with the client machine and the user. Yalcinalp discloses comparing control information associated with the plurality of GUI nodes to credentials associated with the user associated with the request (page 12, lines 1-14). Yalcinalp discloses identifying, based on the user identifier, a subset of the plurality of GUI nodes that the user is authorized to access (Figure 4A, page 10, lines 13-18 and lines 24-28). Based on the client identifier of the requesting user, the tree is searched to determine the set of GUI nodes that are applicable to the client and the URL being requested. Yalcinalp discloses retrieving the subset of GUI nodes the user is authorized to access (page 10, lines 14-18 and Figure 4A). The GUI nodes and the information in the nodes are retrieved based on the identifier of the client. Yalcinalp discloses

Art Unit: 2173

determining a data representation specification associated with the user, the data representation specification specifying one or more data formats appropriate for the user (page 6, lines 5-11 and page 10, lines 14-18). The XSL stylesheet is a determined based on the client identifier and specifies distinct layout and format for the data for a specific client. Yalcinalp discloses converting the subset of GUI nodes the user is authorized to access, conforming to the data representation specification associated with the user (page 10, lines 14-19). The XML data in the GUI nodes are accessed and converted based on the XSL stylesheet that is associated with a client. Yalcinalp does not disclose a script and transmitting the script to the user. Sahota discloses a script that is used for converting of web GUI elements into a format that is appropriate for a client. Sahota disclose transmitting the script to the user. See page 8, paragraph 84. The conversion data is converted to a script and transmitted to the user for providing the converted customized display. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota a script and transmitting the script to the user. Both Yalcinalp and Sahota disclose conversion process for dynamically constructing web pages for various clients. Sahota further discloses providing conversion methods for various different client devices and formats. Sahota discloses multiple ways of conducting this conversion method. This provides motivation for Yalcinalp to learn these additional means of conversion using scripts. Therefore one skilled in the art at the time of the invention would have been motivated to learn from Sahota a script and transmitting the script to the user.

Referring to claim 18, Yalcinalp discloses that the subset of GUI nodes comprise disparate data formats and further comprising converting the subset of GUI nodes into a common data format (Figure 4A, page 10, lines 14-19). The subset of GUI nodes comprise disparate data formats represented as multiple XML documents and XSL formats. The XML documents and XSL files are used to convert the data within these subset of GUI nodes into a common data format that is the final transformed document to be viewed by the user.

Referring to claim 19, Yalcinalp disclose that the GUI nodes comprise XML files (page 10, lines 14-18), where the GUI nodes comprise URL that identify XML files.

Referring to claim 20, Yalcinalp does not disclose that the subset of GUI nodes comprise disparate XML formats and further comprising converting the subset of GUI nodes into a common XML format. Sahota discloses disparate XML formats and further converting the disparate XML formats into a common XML format (page 3, paragraphs 40, 41 and 42), where the templates are the disparate XML formats that are converted to a standardized data stream represented as a standardized XML format. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format. Yalcinalp discloses that a set of XML documents can be accessed and transformed to generate a final document (page 10, lines 14-18). Each of the XML documents can contain different format data and layout suggesting disparate XML formats. In view of this it would have been obvious for Yalcinalp

Art Unit: 2173

to learn from Sahota. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format.

Referring to claim 21, Yalcinalp discloses that converting the subset of GUI nodes into a common XML format comprises using an XSL data conversion specification file (page 10, lines 14-18). The GUI nodes that are associated with a set of XML documents are converted to a format appropriate for one client which is the common XML format and this uses an XSL stylesheet transformation file.

Referring to claim 22, Yalcinalp discloses that the data representation specification comprises an XSL file (page 10, lines 14-18).

Referring to claim 23, Yalcinalp and Sahota disclose that the script comprises HTML, JavaScript, or Java code (Sahota, page 8, paragraph 84, lines 9-12).

Referring to claim 24, Yalcinalp discloses logic embodied in a computer-readable medium operable, when executed by a computer processor, to perform the steps claimed below (page 8, line 28-page 9, line 25). The logic is stored in a computer readable medium executed by the CPU to carry out the functionality. Yalcinalp discloses receiving a request for a webpage identified by a URL (page 7, lines 6-7). Yalcinalp discloses that the webpage is defined by a plurality of GUI nodes stored in a page registry (Figure 4A and page 10, lines 4-19). Yalcinalp discloses receiving a user identifier representing a user associated with the request (page 7, lines 7-11), where the user identifier includes the client type

Art Unit: 2173

and client configuration associated with the client machine and the user.

Yalcinalp discloses comparing control information associated with the plurality of GUI nodes to credentials associated with the user associated with the request (page 12, lines 1-14). Yalcinalp discloses identifying, based on the user identifier, a subset of the plurality of GUI nodes that the user is authorized to access (Figure 4A, page 10, lines 13-18 and lines 24-28). Based on the client identifier of the requesting user, the tree is searched to determine the set of GUI nodes that are applicable to the client and the URL being requested. Yalcinalp discloses retrieving the subset of GUI nodes the user is authorized to access (page 10, lines 14-18 and Figure 4A). The GUI nodes and the information in the nodes are retrieved based on the identifier of the client. Yalcinalp discloses determining a data representation specification associated with the user, the data representation specification specifying one or more data formats appropriate for the user (page 6, lines 5-11 and page 10, lines 14-18). The XSL stylesheet is determined based on the client identifier and specifies distinct layout and format for the data for a specific client. Yalcinalp discloses converting the subset of GUI nodes the user is authorized to access, conforming to the data representation specification associated with the user (page 10, lines 14-19). The XML data in the GUI nodes are accessed and converted based on the XSL stylesheet that is associated with a client. Yalcinalp does not disclose a script and transmitting the script to the user. Sahota discloses a script that is used for converting of web GUI elements into a format that is appropriate for a client. Sahota disclose transmitting the script to the user. See page 8, paragraph 84. The conversion



Art Unit: 2173

data is converted to a script and transmitted to the user for providing the converted customized display. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota a script and transmitting the script to the user. Both Yalcinalp and Sahota disclose conversion process for dynamically constructing web pages for various clients. Sahota further discloses providing conversion methods for various different client devices and formats. Sahota discloses multiple ways of conducting this conversion method. This provides motivation for Yalcinalp to learn these additional means of conversion using scripts. Therefore one skilled in the art at the time of the invention would have been motivated to learn from Sahota a script and transmitting the script to the user.

Referring to claim 25, Yalcinalp discloses that the subset of GUI nodes comprise disparate data formats and wherein the logic is further operable to perform the steps comprising converting the subset of GUI nodes into a common data format (Figure 4A, page 10, lines 14-19). The subset of GUI nodes comprise disparate data formats represented as multiple XML documents and XSL formats. The XML documents and XSL files are used to convert the data within these subset of GUI nodes into a common data format that is the final transformed document to be viewed by the user.

Referring to claim 26, Yalcinalp discloses that the GUI nodes comprise XML files (page 10, lines 14-18), where the GUI nodes comprise URL that identify XML files.

Referring to claim 27, Yalcinalp does not disclose that the subset of GUI nodes comprise disparate XML formats and wherein the logic is further operable to perform the steps comprising converting the subset of GUI nodes into a common XML format. Sahota discloses disparate XML formats and further converting the disparate XML formats into a common XML format (page 3, paragraphs 40, 41 and 42), where the templates are the disparate XML formats that are converted to a standardized data stream represented as a standardized XML format. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format. Yalcinalp discloses that a set of XML documents can be accessed and transformed to generate a final document (page 10, lines 14-18). Each of the XML documents can contain different format data and layout suggesting disparate XML formats. In view of this it would have been obvious for Yalcinalp to learn from Sahota. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format.

Referring to claim 28, Yalcinalp discloses that converting the subset of GUI nodes into a common XML format comprises using an XSL data conversion specification file (page 10, lines 14-18). The GUI nodes that are associated with a set of XML documents are converted to a format appropriate for one client which is the common XML format and this uses an XSL stylesheet transformation file.

Referring to claim 29, Yalcinalp discloses that the data representation specification comprises an XSL file (page 10, lines 14-18).

Referring to claim 30, Yalcinalp and Sahota disclose that the script comprises HTML, JavaScript, or Java code (Sahota, page 8, paragraph 84, lines 9-12).

Referring to claim 31, Yalcinalp discloses a system for dynamically constructing a webpage (page 6, lines 3-21 and Figure 1). Web pages in a NETSCAPE browser are dynamically created using XSL sheets and XML documents. Yalcinalp discloses a GUI data manager comprising a processor operable to carry out the method below (Figure 3 and page 9, lines 20-25). Yalcinalp discloses receiving a request for a webpage identified by a URL and defined by a plurality of GUI nodes stored in a page registry (page 7, lines 6-7, Figure 4A and page 10, lines 4-19). Yalcinalp discloses receive a user identifier representing a user associated with the request (page 7, lines 7-11), where the user identifier includes the client type and client configuration associated with the client machine and the user. Yalcinalp discloses identifying, based on the user identifier, a subset of the plurality of GUI nodes that the user is authorized to access (Figure 4A, page 10, lines 13-18 and lines 24-28). Based on the client identifier of the requesting user, the tree is searched to determine the set of GUI nodes that are applicable to the client and the URL being requested. Yalcinalp discloses retrieving the subset of GUI nodes the user is authorized to access (page 10, lines 14-18 and Figure 4A). The GUI nodes and the information in the nodes are retrieved based on the identifier of the client. Yalcinalp discloses

Art Unit: 2173

determining a data representation specification associated with the user, the data representation specification specifying one or more data formats appropriate for the user (page 6, lines 5-11 and page 10, lines 14-18). The XSL stylesheet is determined based on the client identifier and specifies distinct layout and format for the data for a specific client. Yalcinalp discloses converting the subset of GUI nodes the user is authorized to access, conforming to the data representation specification associated with the user (page 10, lines 14-19). The XML data in the GUI nodes are accessed and converted based on the XSL stylesheet that is associated with a client. Yalcinalp does not disclose a script. Sahota discloses a script that is used for converting of web GUI elements into a format that is appropriate for a client. See page 8, paragraph 84. The conversion data is converted to a script and transmitted to the user for providing the converted customized display. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota a script. Both Yalcinalp and Sahota disclose conversion process for dynamically constructing web pages for various clients. Sahota further discloses providing conversion methods for various different client devices and formats. Sahota discloses multiple ways of conducting this conversion method. This provides motivation for Yalcinalp to learn these additional means of conversion using scripts. Therefore one skilled in the art at the time of the invention would have been motivated to learn from Sahota a script used in conversion transformations.

Referring to claim 32, Yalcinalp discloses a data format translator comprising a processor operable to convert the subset of GUI nodes retrieved by

Art Unit: 2173

the GUI manager from disparate data formats into a common data format (Figure 3, Figure 4A and page 10, lines 14-19). The XSLT Service converts the subset of GUI nodes that are retrieved by the Application Server. The subset of GUI nodes comprise disparate data formats represented as multiple XML documents and XSL formats. The XML documents and XSL files are used to convert the data within these subset of GUI nodes into a common data format that is the final transformed document to be viewed by the user.

Referring to claim 34, Yalcinalp and Sahota disclose further comprising receiving the request for the webpage and user identifier from the user (Yalcinalp, page 7, lines 6-9). Yalcinalp and Sahota disclose transmitting the request for the webpage and user identifier to the GUI manager (Yalcinalp, Figure 3 and page 7, lines 6-9), where the request and client identifier information is processed in the Application Server. Yalcinalp and Sahota disclose receiving the script conforming to the data representation specification associated with the user from the GUI manager (Yalcinalp, page 10, lines 14-19 and Sahota, page 8, paragraph 84) and transmitting the script conforming to the data representation specification association to the user (Sahota, page 8, paragraph 84). Based on the combination disclosed in claim 31, Yalcinalp and Sahota disclose receiving the script conforming to the data representation specification associated with the user from the GUI and transmitting the script conforming to the data representation specification association to the user. Yalcinalp and Sahota disclose a web server operable to perform the steps above (Sahota, page 7, paragraph 74).

Referring to claim 35, Yalcinalp discloses that the GUI nodes comprise XML files (page 10, lines 14-18), where the GUI nodes comprise URL that identify XML files.

Referring to claim 36, Yalcinalp does not disclose that the subset of GUI nodes comprise disparate XML formats and wherein the GUI data manager is further operable to convert the subset of GUI nodes into a common XML format. Sahota discloses disparate XML formats and further converting the disparate XML formats into a common XML format (page 3, paragraphs 40, 41 and 42), where the templates are the disparate XML formats that are converted to a standardized data stream represented as a standardized XML format. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format. Yalcinalp discloses that a set of XML documents can be accessed and transformed to generate a final document (page 10, lines 14-18). Each of the XML documents can contain different format data and layout suggesting disparate XML formats. In view of this it would have been obvious for Yalcinalp to learn from Sahota. It would have been obvious to one skilled in the art at the time of the invention to learn from Sahota disparate XML formats and further converting the disparate XML formats into a common XML format.

Referring to claim 37, Yalcinalp discloses converting the subset of GUI nodes comprises retrieving environmental variables for the user, the environment

Art Unit: 2173

variables comprising one or more private XSL files for the user (page 13, line 28- page 14, line 1).

Referring to claim 38, Yalcinalp discloses converting the subset of GUI nodes comprises retrieving default XSL files, the default XSL files comprising one or more XSL files shared by the user and one or more other users (page 14, lines 1-5).

4. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over Yalcinalp, Sahota and U. S. Patent No. 6,826,597 B1 (Lonnroth et al.), herein referred to as Lonnroth.

Referring to claim 33, Yalcinalp and Sahota disclose a GUI data manager and a page registry storing the plurality of GUI nodes (Yalcinalp, Figure 3 and page 9, lines 20-23). Yalcinalp does not disclose an application integration bus operable to couple the GUI data manager to the page registry. Lonnroth discloses application integration bus that couples a storage system to a process system which accesses data from the storage system (Figure 3 and column 11, lines 6-11). It would have been obvious to one skilled in the art at the time of the invention to learn from Lonnroth an application integration bus operable to couple the GUI data manager to the page registry. As disclosed in Lonnroth, a bus allows for two components to communicate data. This provides motivation for Yalcinalp to have the page registry be coupled to the GUI data manager through a bus, where the data of the page registry can be communicated to the GUI data manager. Therefore, one skilled in the art at the time of the invention would have

Art Unit: 2173

been motivated to learn from Lonnroth an application integration bus operable to couple the GUI data manager to the page registry.

### ***Response to Arguments***

5. Applicant's arguments filed 2/12/09 have been fully considered but they are not persuasive.

Applicant's amendments to claims 31-36 including the addition of the term "processor" is not sufficient to overcome the 35 U.S.C. 101 rejection. The processor defined in the specification can be interpreted as software modules and therefore claims 31-36 are not statutory.

Yalcinalp discloses identifying a subset of GUI nodes that the user is authorized to access. The Figure 4a contains GUI nodes which in response to determining the client and user that is accessing the information determines the GUI nodes that is applicable to this user and therefore authorized for the user to access.

### ***Conclusion***

6. Responses to this action should be submitted as per the options cited below: The United States Patent and Trademark Office requires most patent related correspondence to be: a) faxed to the Central Fax number (571-273-8300) b) hand carried or delivered to the Customer Service Window (located at the Randolph Building, 401 Dulany Street, Alexandria, VA 22314), c) mailed to the mailing address set forth in 37 CFR 1.1 (e.g., P.O. Box 1450, Alexandria, VA 22313-1450), or d) transmitted to the Office using the Office's Electronic Filing System.



Art Unit: 2173

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Namitha Pillai whose telephone number is (571) 272-4054. The examiner can normally be reached from 8:30 AM - 5:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, Kieu Vu can be reached on (571) 272-4057.

All Internet e-mail communications will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirements of 35 U.S.C. 122. This is more clearly set forth in the Interim Internet Usage Policy published in the Official Gazette of the Patent and Trademark on February 25, 1997 at 1195 OG 89.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (571) 272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR

Art Unit: 2173

system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free).

Namitha Pillai  
Patent Examiner  
Art Unit 2173  
May 26, 2009

/Namitha Pillai/

Primary Examiner, Art Unit 2173